



# Shape Optimization of a Submerged Pump for Oil

Séverine Baillet, Jean Brac, Antoine Henrot

## ► To cite this version:

Séverine Baillet, Jean Brac, Antoine Henrot. Shape Optimization of a Submerged Pump for Oil. WCSMO-7 - 7th World Congress on Structural and Multidisciplinary Optimization, May 2007, Séoul, South Korea. hal-00141929

**HAL Id: hal-00141929**

**<https://hal.science/hal-00141929>**

Submitted on 16 Apr 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Shape Optimization of a Submerged Pump for Oil

Séverine BAILLET<sup>†‡</sup>, Jean BRAC<sup>†</sup>, Antoine HENROT<sup>‡</sup>

<sup>†</sup> Institut Français du Pétrole, DTIMA, 1&4 avenue de Bois-Préau, 92852 Rueil-Malmaison Cedex

severine.baillet@ifp.fr, jean.brac@ifp.fr

<sup>‡</sup> Institut Elie Cartan Nancy, Universités Nancy-CNRS-INRIA, BP 239, 54506 Vandœuvre-lès-Nancy Cedex

severine.baillet@iecn.u-nancy.fr, antoine.henrot@iecn.u-nancy.fr

### 1. Abstract

Submerged pumps are helico-axial turbo-machines, composed of a succession of identical stages arranged in series. The aim of a pump is to increase the pressure of the fluid between its inlet and outlet, at a given flow rate. The objective of the shape optimization of such a submerged pump is to minimize the pressure loss per length unit in one stage. The surfaces to re-design are the hub and the blades.

Parametric, 2D B-splines of the third degree are used for the parameterization of the whole 3D geometry. The coordinates of their control points are the control points of the optimization problem. Technical and geometric constraints are expressed as linear and non-linear equality and inequality constraints on the control points' coordinates. The CFD software Fluent is used to solve the Navier-Stokes turbulent equations.

We use an incomplete gradient method to solve the optimization problem. This method is supposed to provide a descent direction for the minimization process. The quality of the incomplete gradient can be measured by checking whether it gives a descent direction or not, and also by comparing it to the result of a finite differences method.

We first give the results of the optimization problem for a single hub, obtained with a classic descent method, with linear equality constraints. Then we present some tests for the validation of these results and, more generally, for the validation of the incomplete gradient method for our problem.

**2. Keywords:** Shape optimization, incomplete gradient, incomplete sensitivities, submerged pump.

### 3. Introduction

At the end of the exploitation of an oil well, surface pumps are no longer efficient enough to extract the oil remaining in the reservoir. Instead of closing the well, oil companies might use submerged pumps, introduced deep into the ground, in order to maintain the production. Those pumps are helico-axial turbo-machines, composed of a succession of identical stages arranged in series. The aim of a pump is to increase the pressure of the fluid between its inlet and outlet, at a given flow rate. The objective of the shape optimization of such a submerged pump is to minimize the pressure loss per length unit in one stage. The surfaces to re-design are the hub and the blades.

Parametric, 2D B-splines of the third degree are used for the parameterization of the whole 3D geometry. The coordinates of their control points are the control points of the optimization problem. Technical and geometric constraints are expressed as linear and non-linear equality and inequality constraints on the coordinates of the control points.

We use the CFD software Fluent to solve the Navier-Stokes equations with turbulence and we can't calculate the exact gradient of the objective function and consequently we use an incomplete gradient method, as follows. Since the objective function chosen is written as the difference between the integrals of the pressure on the stage's inlet and on its outlet, divided by the stage's length, we use a formula of derivation with respect to the domain in order to get the analytical expression of the gradient. Then we only keep in this expression the terms that we are able to compute by numerical or calculating means. This method is supposed to provide a descent direction for the minimization process. The quality of the incomplete gradient can be measured by checking if it actually gives a good direction, and also by comparing it to the result of a finite differences method.

Shape optimization problems have already been considered in the field of turbomachinery and have often been solved thanks to neural network methods and genetic algorithms, see [1], [2], [3]. We cite [4] and [5] for their use of incomplete gradient methods for the optimization of wings in aerodynamics and of blades in turbomachinery. The literature about shape optimization is very large but we cite [6] and [7] for applications more fluid mechanics oriented.

The optimization process of the submerged pump has been divided in three steps. The first one consists in the shape optimization of the pump's hub, the second one in the shape optimization of its blades. Finally the third step is the shape optimization of the whole pump by combination of the two previous codes.

In this paper we present the results of the optimization problem for a single hub, for a given number of control points and of linear equality constraints. First, we expose the results we obtain by solving this problem with the incomplete gradient method. Though it provides a good direction during the first iterations of the process, after a while the method does not give a descent direction anymore. Therefore we implemented a finite differences method in order to compare it to our method and judge of its quality. We give the conclusions of this comparison in this paper. We then finish with some perspectives.

#### 4. Problem modeling

We consider in this paper a geometry that is similar in size with a pump stage. The blades have been suppressed and the hub is fixed. As a consequence, the fluid simply flows between the cylindrical carter and the hub. Let us call this domain  $\Omega$ . We add two extensions before the inlet and after the outlet of  $\Omega$ . An extension is composed of two long cylinders assembled to the carter and to the hub. These extensions are essential for the correct modeling of the flow in  $\Omega$ .

We study the flow of a gasoil-like fluid. The homogeneous, incompressible, stationnary and single-phase Navier-Stokes equations are solved with the CFD software Fluent. We use a realizable  $k - \epsilon$  model. The Reynolds number is approximately  $10^6$ . We impose a no-slip condition on the walls, a velocity condition on the inlet of the first extension and a pressure condition on the outlet of the second extension.

The objective of the shape optimization is to minimize the pressure loss per length unit in the domain  $\Omega$ . Therefore, a natural choice for the objective function is:

$$F = \frac{\int_I p - \int_O p}{L} \quad (1)$$

where  $p$  is the pressure,  $I$ ,  $O$  and  $L$  are the inlet, outlet and length of  $\Omega$ .

We choose parametric 2D B-splines of the third degree for the modeling of the geometry. These are  $\mathcal{C}^2$ -curves and the shift of a control point only entails a local deformation of the curve, which is convenient in the framework of a shape optimization. A B-spline is a linear combination of the following base functions, defined for  $t \in [0, 1]$ :

$$\begin{cases} p_0(t) = (t^3)/6, \\ p_1(t) = (-3t^3 + 3t^2 + 3t + 1)/6, \\ p_2(t) = (3t^3 - 6t^2 + 4)/6, \\ p_3(t) = (-t^3 + 3t^2 - 3t + 1)/6. \end{cases} \quad (2)$$

The hub is a surface of revolution generated by a B-spline around the  $0x$  axis. Let us denote by  $n \in \mathbb{N}$  the number of control points of the B-spline. Let  $(x_i, y_i)_{i=1\dots n}$ , a set of  $n$  points of  $\mathbb{R}^2$ , be these control points. Then the B-spline can be considered as the collection of  $(n - 3)$  B-spline segments  $(x_i(t), y_i(t))$  defined by:

$$\forall i \in \{1, \dots, n - 3\}, \begin{cases} x_i(t) = \sum_{j=0}^3 x_{i+j} p_{3-j}(t), \\ y_i(t) = \sum_{j=0}^3 y_{i+j} p_{3-j}(t), \end{cases} \text{ for } t \in [0, 1]. \quad (3)$$

The extensions are not deformed during the optimization process. As a first consequence,  $L$  is constant and the derivatives of the denominator of the objective function are null (we will come back to this later). As a second consequence, a constraint has to be written to ensure the  $\mathcal{C}^0$  continuity at the interface between the central cylinder of the extensions and the hub. Let  $R$  be the radius of the central cylinders. Then the  $\mathcal{C}^0$  continuity equality constraints are written at the inlet interface as:

$$x_1 p_3(0) + x_2 p_2(0) + x_3 p_1(0) + x_4 p_0(0) = 0, \quad (4)$$

$$y_1 p_3(0) + y_2 p_2(0) + y_3 p_1(0) + y_4 p_0(0) = R, \quad (5)$$

and, at the outlet interface, as:

$$x_{n-3}p_3(1) + x_{n-2}p_2(1) + x_{n-1}p_1(1) + x_n p_0(1) = L, \quad (6)$$

$$y_{n-3}p_3(1) + y_{n-2}p_2(1) + y_{n-1}p_1(1) + y_n p_0(1) = R. \quad (7)$$

These first equality constraints are necessary. Some other constraints can also be imposed, like  $\mathcal{C}^1$  continuity constraints that are written as:

$$y_1 p'_3(0) + y_2 p'_2(0) + y_3 p'_1(0) + y_4 p'_0(0) = 0, \quad (8)$$

$$y_{n-3} p'_3(1) + y_{n-2} p'_2(1) + y_{n-1} p'_1(1) + y_n p'_0(1) = 0. \quad (9)$$

## 5. Incomplete gradient computation

The CFD software Fluent has historically been used for the modeling of the submerged pumps at IFP. With the will to keep this background accessible and up to date, the decision was taken to lead a shape optimization coupled with this same CFD software. The drawback of this approach is that no exact gradient calculation is possible. On the other hand, the software provides us with the values of the state functions (pressure, velocity,...) and of their derivatives on the boundary of the domain. Considering this, the idea came to us to compute an approximate gradient of the objective function and then to lead the optimization with a gradient-based method.

As we said before, the denominator of the objective function being a constant, its derivatives are null. As a consequence, we only have to calculate the derivatives of the numerator of  $F$ , that we will call  $J$ . Let us first introduce some notations.  $\Omega$  is the flow domain, delimited by the hub  $D$ , the carter  $C$ , the inlet  $I$  and the outlet  $O$ . Let  $\partial\Omega$  be the boundary of  $\Omega$ . Then

$$\partial\Omega = D \cup C \cup I \cup O. \quad (10)$$

The hub  $D$  is the only face that is deformed during the optimization, while the other faces are not. We can transform the expression of  $J$ :

$$J = \int_O p - \int_I p = \int_{I \cup O} p n_x = \int_{\partial\Omega} p n_x - \int_D p n_x - \int_C p n_x, \quad (11)$$

where  $n$  is the normal vector, pointing out of the flow domain  $\Omega_h$ . But, as the carter is a cylinder with axis  $0x$ , the final expression of  $J$  is:

$$J = \int_{\partial\Omega} p n_x - \int_D p n_x := J_1 - J_2. \quad (12)$$

We hereby see that it will be necessary to calculate derivatives with respect to the domain in order to compute  $J'$ .

For this purpose we use the following results taken from [8]. Let  $h \in [0, h_M[ \rightarrow \Phi(h)$  be a family of  $\mathcal{C}^1$ -diffeomorphisms of  $\mathbb{R}^N$  such that:  $\Phi : h \in [0, h_M[ \rightarrow W^{1,\infty}(\mathbb{R}^N)$  is differentiable at 0 with  $\Phi(0) = Id$  and  $\Phi'(0) = \mathcal{V}$ . We define  $\Omega_h = \Phi(h)(\Omega)$ , where  $\Omega$  is a given open set at least  $\mathcal{C}^1$ . Let  $n_h$  be the normalized normal vector to the boundary  $\partial\Omega_h$ , pointing out of the flow domain  $\Omega_h$ . We want to compute the derivative of such an expression:  $h \rightarrow G(h) = \int_{\partial\Omega_h} g(h)$  where  $g(h) : \partial\Omega_h \rightarrow \mathbb{R}$  is given.

Supposing that there is enough regularity for  $\Omega$ ,  $g$  and the  $\Phi(h)$  family, the following theorem can be proved:

**Theorem 1.** *Let  $\mathcal{N}$  be a normalized  $\mathcal{C}^1$  extension to  $\mathbb{R}^N$  of  $n$ . Let us suppose that  $\Omega$  is  $\mathcal{C}^3$ , that  $h \rightarrow \Phi(h) \in \mathcal{C}^2$  is differentiable at 0 with  $\Phi(0) = Id$ ,  $\Phi'(0) = \mathcal{V}$ , and that  $h \rightarrow g(h) \in W^{1,1}(\mathbb{R}^N)$  is differentiable at 0 with  $g(0) \in W^{2,1}(\mathbb{R}^N)$ . Then,  $h \rightarrow G(h)$  is differentiable at 0 and, if  $g'(0) := \frac{\partial}{\partial h}|_{h=0} g(h)$ , we have that:*

$$G'(0) = \int_{\partial\Omega} (g'(0) + \nabla g(0) \cdot \mathcal{N} \mathcal{V} \cdot \mathcal{N} + g(0) H \mathcal{V} \cdot \mathcal{N}) \quad (13)$$

where  $H = \text{div}(\mathcal{N})$  is the mean curvature of  $\partial\Omega$ .

Generally, the derivatives with respect to the deformation of the  $g'$  kind are the terms that we will come to omit in the expression of our gradient. Thus, the problem of the incomplete gradient method is that we are compelled to suppress some information that we are not able to retrieve. As  $J$  in Eq.(12) is only a function of the pressure  $p$ , we try to produce some new information by introducing the velocity  $U = (u_x, u_y, u_z)$  thanks to the Navier-Stokes equation, like this:

$$J_1 = \int_{\partial\Omega} p n_x = \int_{\Omega} \frac{\partial p}{\partial x} = \int_{\Omega} (\mu \Delta u_x - \rho U \cdot \nabla u_x). \quad (14)$$

Using the Green formula we get:

$$J_1 = \mu \int_{\partial\Omega} \frac{\partial u_x}{\partial n} - \rho \int_{\partial\Omega} u_x U \cdot n. \quad (15)$$

Using the decomposition Eq.(10) of  $\partial\Omega$  and the formula Eq.(13) of Theorem 1., we calculate  $J'_1$ . We omit the terms as  $U'$ ,  $u'_x$  and  $(\nabla u_x)'$  and finally get:

$$J'_1 \simeq \mu \left( \int_D \nabla u_x \cdot \mathcal{N}' + \int_D \nabla u_x \cdot \mathcal{N} H \mathcal{V} \cdot \mathcal{N} \right). \quad (16)$$

We apply formula Eq.(13) directly to  $J_2$  and, omitting  $p'$ , we get:

$$J'_2 = \left( \int_D p n_x \right)' \simeq \int_D p \mathcal{N}'_x + \int_D \mathcal{N}_x \nabla p \cdot \mathcal{N} \mathcal{V} \cdot \mathcal{N} + \int_D p \mathcal{N}_x H \mathcal{V} \cdot \mathcal{N}. \quad (17)$$

To conclude, the final expression of the incomplete gradient is:

$$J' = \mu \left( \int_D \nabla u_x \cdot \mathcal{N}' + \int_D \nabla u_x \cdot \mathcal{N} H \mathcal{V} \cdot \mathcal{N} \right) - \left( \int_D p \mathcal{N}'_x + \int_D \mathcal{N}_x \nabla p \cdot \mathcal{N} \mathcal{V} \cdot \mathcal{N} + \int_D p \mathcal{N}_x H \mathcal{V} \cdot \mathcal{N} \right). \quad (18)$$

In this expression, the terms  $u_x$ ,  $\nabla u_x$ ,  $p$  and  $\nabla p$  can be retrieved thanks to the CFD software. The terms  $\mathcal{V}$ ,  $\mathcal{N}$ ,  $\mathcal{N}'$  and  $H$  can be calculated as we will now see.

The shift of a control point of the B-spline entails a deformation of the geometry that can be represented by a vector field. This vector field actually is the  $\mathcal{V}$  vector we need to calculate. Let us consider the simple case where the B-spline has a single segment  $(x_1(t), y_1(t))$ , generated by four control points  $(x_i, y_i)_{i=1\dots 4}$ , and where we shift the first control point:  $(x'_1, y'_1) = (x_1, y_1) + \delta(v_x, v_y)$ . The single segment  $(x'_1(t), y'_1(t))$  of the new B-spline is such that:

$$\forall t \in [0, 1], \begin{cases} x'_1(t) = \sum_{j=0}^3 x'_{1+j} p_{3-j}(t) = \sum_{j=0}^3 x_{1+j} p_{3-j}(t) + \delta v_x p_3(t) = x_1(t) + \delta v_x p_3(t), \\ y'_1(t) = \sum_{j=0}^3 y'_{1+j} p_{3-j}(t) = \sum_{j=0}^3 y_{1+j} p_{3-j}(t) + \delta v_y p_3(t) = y_1(t) + \delta v_y p_3(t). \end{cases} \quad (19)$$

Thus, we have:

$$\forall t \in [0, 1], (x'_1(t), y'_1(t)) = (x_1(t), y_1(t)) + \delta V(t), \quad (20)$$

where  $V$  is the deformation vector field on the B-spline:

$$\forall t \in [0, 1], V(t) = p_3(t)(v_x, v_y). \quad (21)$$

This simple case can easily be extended to cases with more control points and with any control points being shifted.

In order to use formula Eq.(13), we then define an extension  $\mathcal{V}$  of  $V$  to  $\mathbb{R}^2$ , as well as an extension  $\mathcal{N}$  of

the normal vector  $n$ . We do not develop this part here but we give the following interesting intermediate result:

$$\mathcal{N}' = -{}^t D\mathcal{V}\mathcal{N} - D\mathcal{N}\mathcal{V} = -\nabla(\mathcal{V}\mathcal{N}). \quad (22)$$

Using differential geometry we can finally calculate  $\mathcal{N}'$ , but also  $H$  and have thus recovered all the terms required in the expression Eq.(18).

## 6. Numerical results

### 6.1. Problem discussion and optimization method

The modeling of the problem is done as in section 4. for a hub represented by a single segment B-spline. The initial B-spline is build so that its characteristic lengths are of the same order as those of the hub of the pump to be improved.

The single segment B-spline is generated by four control points and each of them has two coordinates. Therefore there are eight control points for the optimization problem. We impose four equality constraints (Eq.(4) to Eq.(7)) to ensure the  $\mathcal{C}^0$  continuity at the interface between the hub and the extensions. Thus the problem has four degrees of freedom.

We recall that the objective of the optimization is to minimize the objective function:

$$F = \frac{\int_I p - \int_O p}{L}.$$

The gradient of  $F$  is obtained by computing the expression Eq.(18). For the optimization we use a classic descent method. In order to cope with the imperfections of the incomplete gradient, the step length is chosen to be a small constant.

As the equality constraints are linear, they can be taken into account by reducing the problem. The first and fourth B-spline control points are expressed as functions of the second and third control points thanks to the equalities. We then compute a reduced gradient of the objective function and lead the optimization based on this gradient.

### 6.2. Results of the incomplete gradient method

The pressure loss per length unit, with respect to the iterations, is displayed on Fig.1. The objective function  $F$  decreases correctly during the 16 first iterations but increases afterwards.

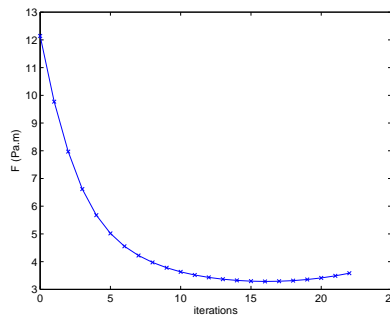


Figure 1: Objective function with respect to the iterations, with the incomplete gradient method.

On Fig.2 we represent the evolution of the shape of the B-spline that generates the hub, in the  $(x, y)$  reference frame, with respect to the iterations. The initial B-spline is the top black curve. The successive B-splines are colored from black to blue and then to green. The curve that corresponds to the lowest value obtained for  $F$  (at the 16th iteration) is displayed in red.

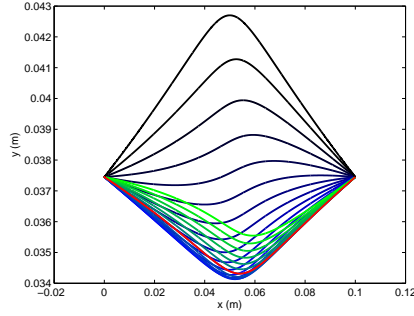


Figure 2: Evolution of the shape of the hub, with the incomplete gradient method.

The fluid flows between the hub and the cylindrical carter. As we want to minimize the pressure loss, we expected that a cylindrical shape for the hub would be optimal. We can now see that the pressure loss is even smaller when the hub gets a little narrower than a cylinder.

We can conclude from these first observations that the incomplete gradient is not a very good approximation of the gradient of the objective function  $F$ . Therefore we will now present some ideas to estimate a little more precisely the quality of the method.

### 6.3. Validation of the method

The first way to estimate the quality of the incomplete gradient method is, obviously, to make sure that it gives a descent direction through the iterations. We have just seen that, though it is the case during the first iterations, it is not the case anymore at the end of the process.

We can compare the results of the incomplete gradient method to the results of a finite differences method. On Fig.3 we represent the evolution of the shape of the B-spline that generates the hub, in the  $(x, y)$  reference frame, with respect to the iterations. Once again, the initial B-spline is the top black curve and the successive B-splines are colored from black to blue and finally to green for the last iteration.

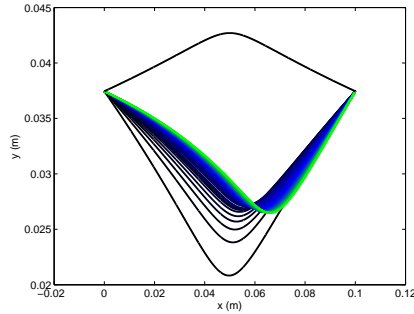


Figure 3: Evolution of the shape of the hub, with the finite differences method.

We can see that with both methods the hub tends to get narrower than a cylinder, and this, closer to its outlet than to its inlet. However, on Fig.4 we can see that the hub obtained thanks to the finite differences method is way more narrow than the one obtained thanks to the incomplete gradient method.

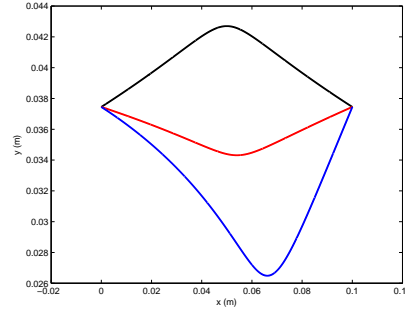


Figure 4: Comparison of the best shape obtained with the incomplete gradient method, in red, with the final shape obtained with the finite differences method, in blue. The initial geometry is recalled in black.

Comparing on Fig.5 the evolution of the objective function, with respect to the iterations, confirms that the pressure loss per length unit is more reduced by the finite differences method than by the incomplete gradient method.

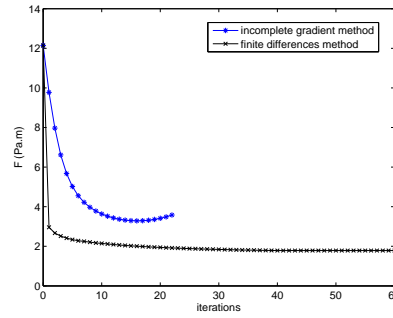


Figure 5: Objective function with respect to the iterations.

Finally we can compute the angle between the direction of the incomplete gradient and the direction of the gradient computed with the finite differences. We see that the angle is getting close to zero during the first iterations, but is then increasing during the following iterations. It remains smaller than  $\pi/2$  until the 16th iteration and then becomes greater than  $\pi/2$  from the moment the incomplete gradient does not give a descent direction anymore.

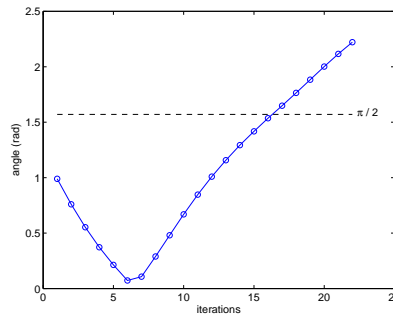


Figure 6: Angle between the direction of the incomplete gradient and the direction of the gradient computed with the finite differences, with respect to the iterations.



Though the finite differences method gives good results, it should be noticed that it is very costly. Indeed, if there are  $k$  degrees of freedom to the optimization problem, then it asks for  $k + 1$  evaluations of the cost function, that is to say  $k + 1$  Navier-Stokes resolutions. Therefore this method can only be considered as a tool for the validation of another method, in the case of a small number of degrees of freedom.

A last way we imagined to judge of the quality of the incomplete gradient is to estimate the omitted terme  $p'$ . It is well known that, when using incomplete gradients, the normal gradient of the pressure should be small. This is because  $p' \sim \nabla p \cdot n V \cdot n$  on the deformed surfaces. On Fig.7 we can see that the norm of the approximation  $\int_D \nabla p \cdot n n_x V \cdot n$  of the omitted  $\int_D p' n_x$  term is quite small in comparison with the norm of the incomplete gradient.

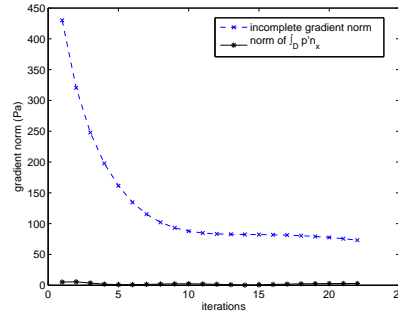


Figure 7: Norm of the incomplete gradient and norm of the approximation term of  $\int_D p' n_x$  with respect to the iteration of the incomplete gradient method.

## 7. Conclusions

In this paper we presented some results for the shape optimization of the hub of a submerged pump for oil. Parametric, 2D B-splines were chosen as convenient tools for the parameterization of the geometry. The CFD software Fluent was used for the resolution of the turbulent Navier-Stokes equations. We implemented an incomplete gradient method for the minimization of the pressure loss per length unit in the geometry.

The results that are presented in this paper show that the method is not so good. In a first time it gives a descent direction but afterwards this is not the case anymore. The finite differences method gives better results, even if the general behaviour of the pressure loss decrease and of the deformation of the shape with the iterations are in agreement in the two methods.

To go further, we could try to implement more efficient methods for the optimization (quasi-Newton,...). In the framework of the optimization of the whole pump, the analytic formulation and the numerical implementation of (in-)equality (non-)linear constraints appear to be crucial. Methods and codes that would allow us to take these constraints into account should be tested.

## 8. References

- [1] A. Demeulenaere, R.A. Van den Braembussche, Three-dimensional inverse method for turbomachinery blading design, *ASME Journal of turbomachinery*, vol. 120, 1998.
- [2] R.A. Van den Braembussche, Turbomachinery blades design systems, *VKI, Lecture Series 1999-02*.
- [3] S. Pierret, Designing turbomachinery blades by means of the function approximation based on artificial neural network, genetic algorithm and the Navier Stokes equations, *PhD thesis, Faculté Polytechnique de Mons (Belgique) and VKI*, 1999.

- [4] M. Stanciu, Gradient incomplet pour l'optimisation de formes en aérodynamique. Applications aux turbomachines, *PhD thesis, Université de Montpellier II, Sciences et Techniques du Languedoc*, 2001.
- [5] E. Leclerc, Contrôle sub-optimal pour les écoulements instationnaires turbulents, *PhD thesis, Université des Sciences et Techniques du Languedoc*, 2003.
- [6] E. Laporte, P. Le Tallec, Numerical Methods in Sensitivity Analysis and Shape Optimization, *Birkhäuser, Boston*, 2003.
- [7] B. Mohammadi, O. Pironneau, Applied Shape Optimization for Fluids, *Clarendon Press, Oxford*, 2001.
- [8] A. Henrot, M. Pierre, Variation et Optimisation de Formes, *Springer, Berlin/Heidelberg/New York*, 2005.